NED UNIVERSITY
OF ENGINEERING AND TECHNOLOGY

Course/Course Code:

Data Acquisition and Microcontroller (MC-546)

Lecture # 07

Course Teacher:

Dr. Tariq Rehman

Universal Serial Bus (USB)

- The **universal serial bus (USB)** is a widely used standard serial bus for connecting peripherals to a computer.
- There are typically two or more USB ports on computers and, with USB hubs, up to 127 devices can be connected.
- USB allows the devices to be connected or disconnected while the computer is running (hot swapping).
- Figure 1 shows the symbol for USB.
- The original USB standard was 1.0, which was followed by 1.1. USB 2.0 replaced the two original versions and more recently USB 3.0 was introduced.
- The earlier versions are still in use, especially 2.0. In terms of data rate, USB has four classifications: low-speed, full-speed, high-speed, and super-speed.

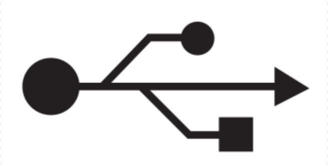
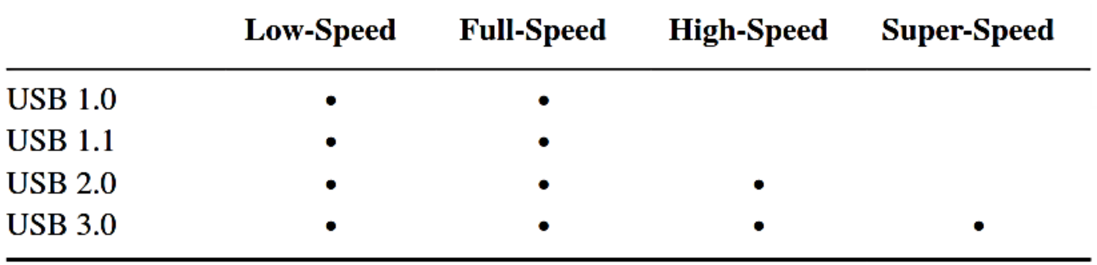


FIGURE 1: USB symbol

Universal Serial Bus (USB)

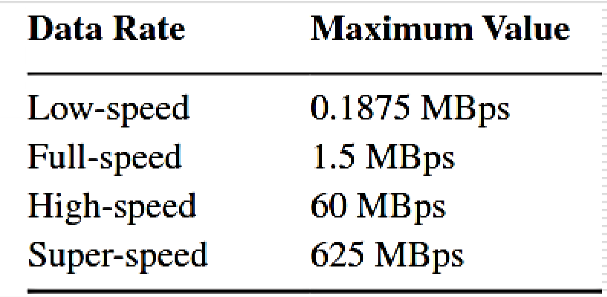
- Table 1 shows how the data rate classifications apply to each of the USB versions and Table 2 shows the data rate values.
- Cable length is an important specification for buses.
- Table 3 lists maximum cable lengths for three USB versions and maximum total lengths when multiple cables are strung together using USB hubs.
- A hub is a common connection device with multiple ports.

TABLE 1:



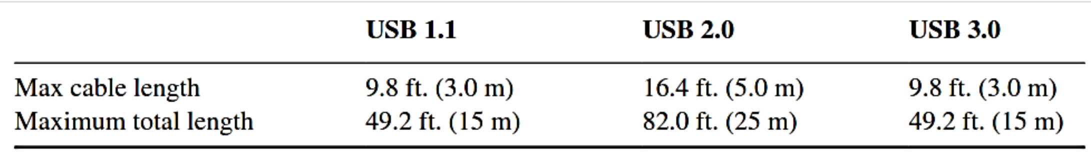
	Low-Speed	Full-Speed	High-Speed	Super-Speed
USB 1.0	•	•		
USB 1.1	•	•		
USB 2.0	•	•	•	
USB 3.0	•	•	•	•

TABLE 2:



Data Rate	Maximum Value
Low-speed	0.1875 MBps
Full-speed	1.5 MBps
High-speed	60 MBps
Super-speed	625 MBps

TABLE 3:



	USB 1.1	USB 2.0	USB 3.0
Max cable length	9.8 ft. (3.0 m)	16.4 ft. (5.0 m)	9.8 ft. (3.0 m)
Maximum total length	49.2 ft. (15 m)	82.0 ft. (25 m)	49.2 ft. (15 m)

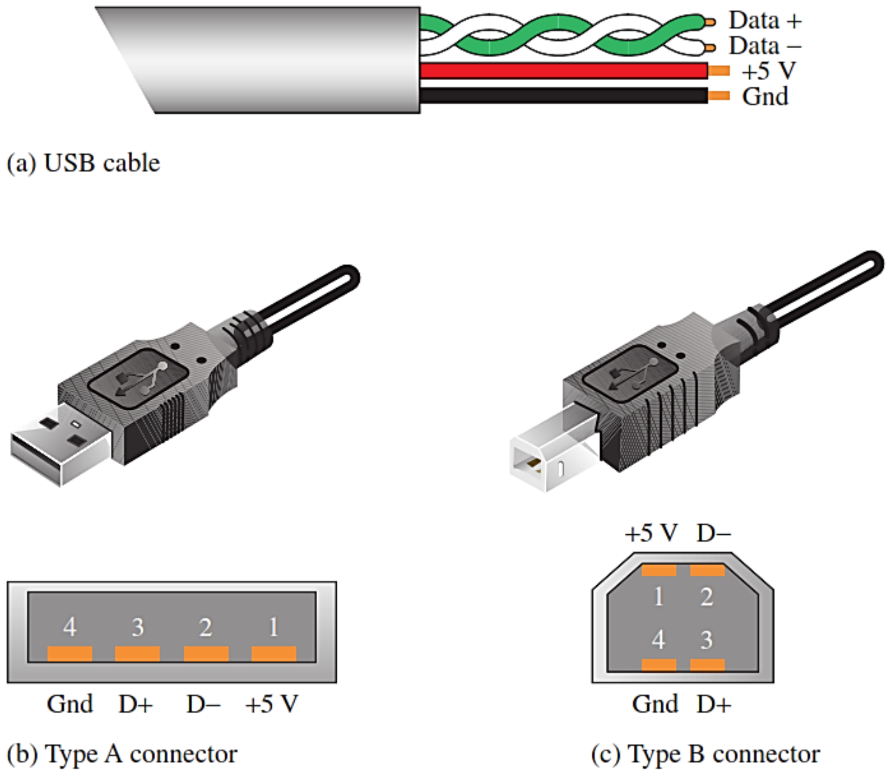
USB Cable and Connectors

☐ USB versions up to and including 2.0 have a four-wire cable that includes a twisted pair to reduce or eliminate noise for data transmission, a +5 V wire, and a ground wire color-coded, as shown in Figure 2(a).

☐ The standard type A and type B connectors are shown in parts (b) and (c) with pin designations.

☐ USB hosts (computer) and devices (peripherals) have sockets, and all USB cables have a type A plug at one end and a type B plug at the other.

☐ The sockets on a host are Type A, and the sockets on peripheral devices are Type B. Hubs have both Type A and Type B. The USB standard also specifies smaller connectors designated mini and micro.



(a) USB cable

(b) Type A connector

(c) Type B connector

FIGURE 2: USB cable and connectors for USB standards through 2.0.

USB Data Format

- Serial data are transmitted on the twisted pair ($D+$ and $D-$) using half-duplex differential mode to minimize EMI and improve the signal-to-noise ratio.
- Data are sent in packets using NRZI (non-return-to-zero invert) encoding format with a 3.3 V level (differentially, there are 6.6 V between the two data lines).

A packet format can contain the following fields:

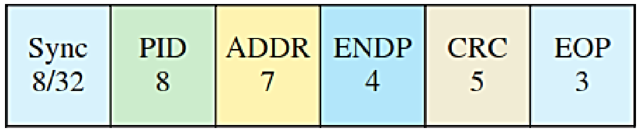
- **Sync field:** All packets start with a sync (synchronization) field. The sync field consists of 8 bits for low and 32 bits (full speed) for high speed and is used to synchronize the receiver clock with that of the transmitter.
- **PID field:** The packet identification field is used to identify the type of packet that is being transmitted. There are four bits in the PID; however, to ensure it is received correctly, the four bits are complemented and repeated, making an 8-bit PID code.

USB Data Format

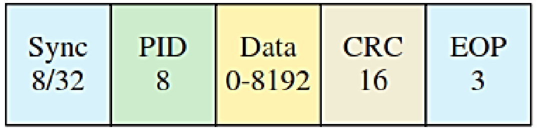
- **ADDR field:** The address field specifies to which device the packet is sent. The seven bits in this field allow for 127 devices to be supported. Address 0 is invalid.
- **Data field:** The data field contains up to 1024 bytes of data.
- **ENDP field:** The endpoint field is made up of four bits, allowing 16 possible end- points. An endpoint is a data source or load. Low-speed devices, however, can only have two additional endpoints on top of the default pipe (four endpoints max). Endpoints can be described as sources or sinks of data.
- **CRC field:** Cyclic redundancy checks are performed on the data within the packet using from 5–16 bits, depending on the type of packet.
- **EOP field:** This packet field signals the end of a packet.

USB Data Format

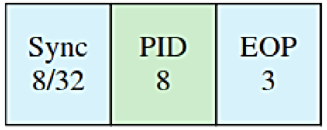
- Four types of USB packets are token, data, handshake, and start-of-frame, as shown in Figure 3 with the packet format for each type.
- Each field is labeled and the number of bits shown.
- The token packet indicates the type of transaction, the data packet contains the actual data, the handshake packet acknowledges a transaction, and the start-of-frame packet begins a new frame.
- The token packet, data packet, hand- shake packet, and start-of-frame packet each have a different packet format as specified by the PID field.



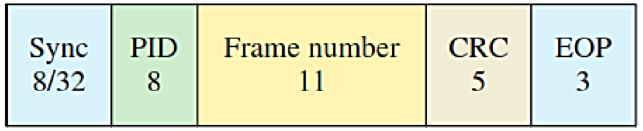
(a) Token packet



(b) Data packet



(c) Handshake packet

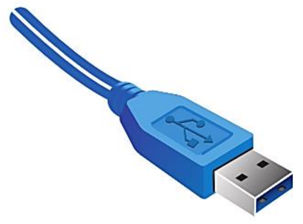


(d) Start-of-frame packet

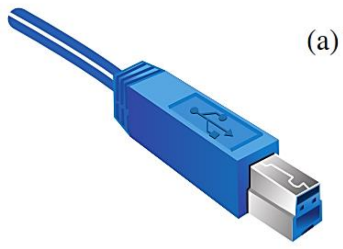
FIGURE 3: Types of USB packets

USB 3.0

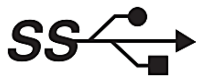
- USB 3.0 is a recent version of the USB standard. The 3.0 version, known as SuperSpeed USB, is ten times faster than USB 2.0 at 4.8 Gbps.
- The 3.0 connectors (types A and B) are different then version 2.0 because they now have nine contacts instead of four.
- The 3.0 type A connector looks about the same as that for 2.0 except that the extra five pins are further inside and make contact only with a 3.0 mating connector.
- The connector is compatible with a 2.0 device where the front four pins are accessible.
- Type A and type B connectors for USB 3.0 along with the standard symbol are shown in Figure 4. There is also a micro-B connector available.



(a) Type A



(b) Type B



(c) Symbol

FIGURE 4: USB 3.0 connectors and symbol

USB 3.0

- USB 3.0 is, for the most part, backward compatible with USB 2.0, but the speed is limited to the 2.0 specification.
- The USB 3.0 cable consists of two additional twisted pairs for data and an additional ground, for a total of nine wires.
- Unlike the previous versions, version 3.0 is full-duplex, meaning that data can be sent and received simultaneously.
- One twisted pair is for receiving data, and two additional twisted pairs are for sending high- speed data.
- The recent USB 3.1 specification increases the data rate to 10 Gbps, twice that of the 3.0.
- USB 3.1 is backward compatible with 3.0 and 2.0 and a new connector, type C, is included in the specification.

USB 3.0

- ☐ Figure 5 shows the USB in a typical computer system. The computer acts as a host and uses Type A connectors. The hub functions as both a host and a device.

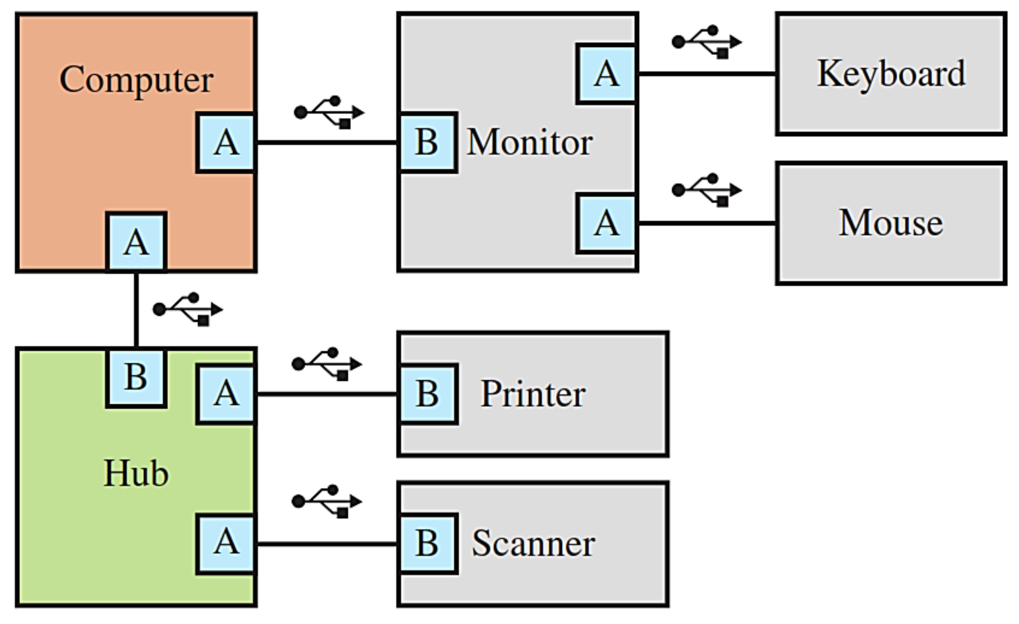


FIGURE 5: Example of USB applications

RS-232 Bus

- Also known as EIA-232, the RS-232 bus was once standard on computers for connection to peripheral devices.
- The standard provides for single-ended data transmission in either synchronous or asynchronous formats.
- It has been replaced by the USB because of its limited speed, relatively large voltage requirements, and large connector size.
- However, RS-232 devices are still used in industrial and telecommunication applications as well as scientific instrumentation.
- The devices connected by the RS-232 are classified as DTE (data terminal equipment) or DCE (data communication equipment).
- Since newer computers have no RS-232 ports, USB-to-RS-232 converters can be used to connect to older RS-232 compatible peripherals, if necessary.

RS-232 Bus

- The standard is designed for one transmitting device and one receiving device with a maximum cable length of 50 feet between them.
- The maximum RS-232 data rate is 20 kbps.
- The data format typically consists of seven or eight bits of data, a start bit, a parity bit in some cases depending on the protocol, and a stop bit.
- A transmitted signal level between +5 V and +15 V represents a binary 0 and between -5 V and -15 V represents a binary 1.
- The data is transmitted in NRZ format, as Figure 6 shows.
- The standard 25-pin connector for RS-232 is shown in Figure 7(a). A smaller 9-pin connector, is shown in part (b).

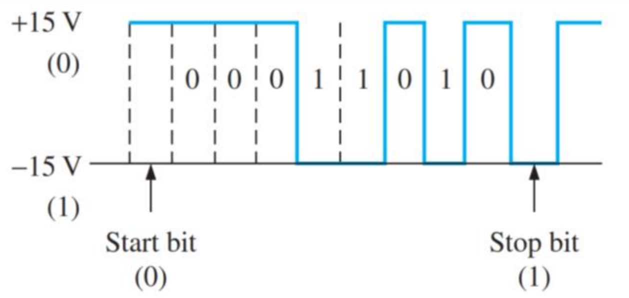


FIGURE 6: Example of RS-232 transmitted data format.

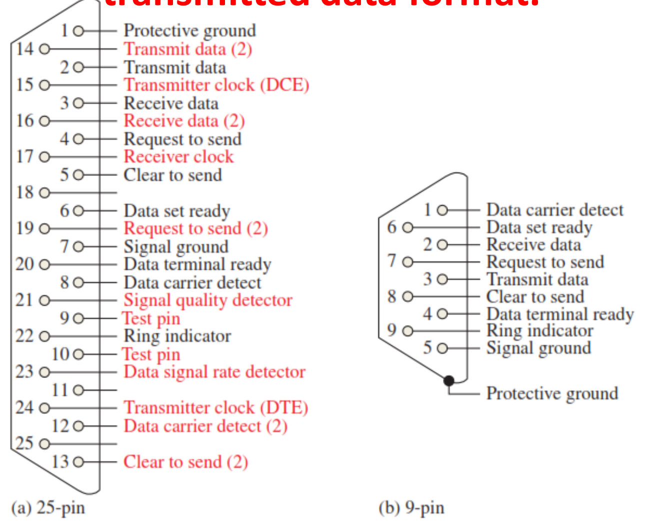
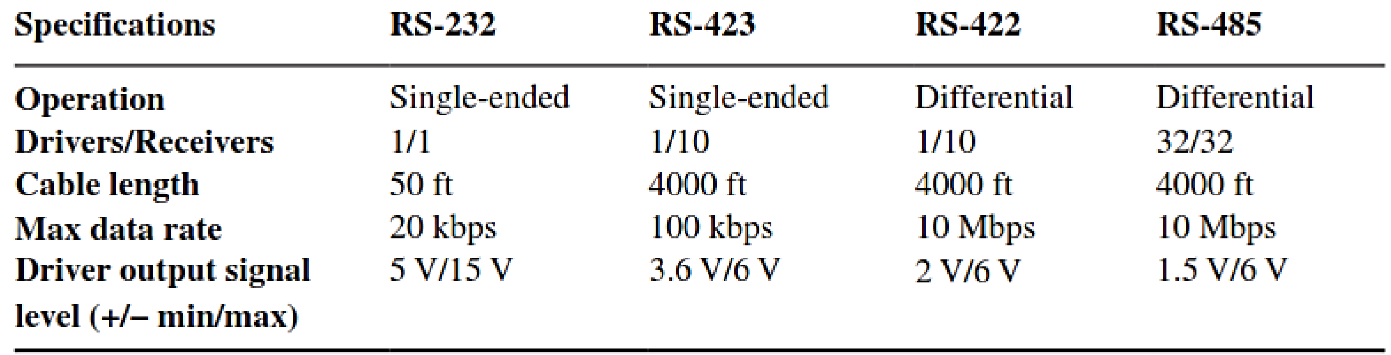


FIGURE 7: Standard RS-232 connectors

RS-422/423/485 Bus

- The RS-422 bus provides for differential transmission for greater distances (longer cable length) and has higher data rates than the RS-232 bus.
- Also, the standard defines the number of receiving devices as ten for a line with one driver (transmitting device) compared to one receiving device for the RS-232.
- The RS-423 bus is similar to the RS-232 in that it is single-ended, but it has a higher data rate and longer cable length.
- The RS-485 bus is a differential standard and can accommodate multiple drivers and receivers.
- Table 4 summarizes some of the features of the RS (EIA) buses.

TABLE 4: Standard RS-232 connectors.



Specifications	RS-232	RS-423	RS-422	RS-485
Operation	Single-ended	Single-ended	Differential	Differential
Drivers/Receivers	1/1	1/10	1/10	32/32
Cable length	50 ft	4000 ft	4000 ft	4000 ft
Max data rate	20 kbps	100 kbps	10 Mbps	10 Mbps
Driver output signal level (+/– min/max)	5 V/15 V	3.6 V/6 V	2 V/6 V	1.5 V/6 V

The SPI Bus

- The **serial-to-peripheral interface** (SPI) bus is a synchronous serial communications bus that uses four wires for communication between a “master” device and a “slave” device.
- This standard was developed by Motorola; it operates in full-duplex mode up to a data rate of 10 Mbps and can accommodate multiple slaves.
- The four signal wires are
    1. MOSI (master out slave in) is initiated by the master and received by the slave.
    2. MISO (master in slave out) is initiated by the slave and received by the master.
    3. SCLK (serial clock) is generated by the master for synchronizing data transfers.
    4. SS (slave select) is generated by the master to select an individual slave.
- Other names are sometimes assigned to these signals such as SDI (serial data in) for MOSI and SDO (serial data out) for MISO. Figure 8 shows a master with a single slave.

The SPI Bus

- The SPI bus is typically used in embedded systems and on PCBs for communication between microprocessors or microcontrollers and peripheral IC chips or between two processors.
- SPI applications include digital audio, signal processing, and telecommunications. SPI is used to communicate with various types of peripherals such as sensors, camera lenses, flash memory, LCD displays, and video games.
- Many microprocessors and microcontrollers include SPI controllers that can be used as either a master or a slave.

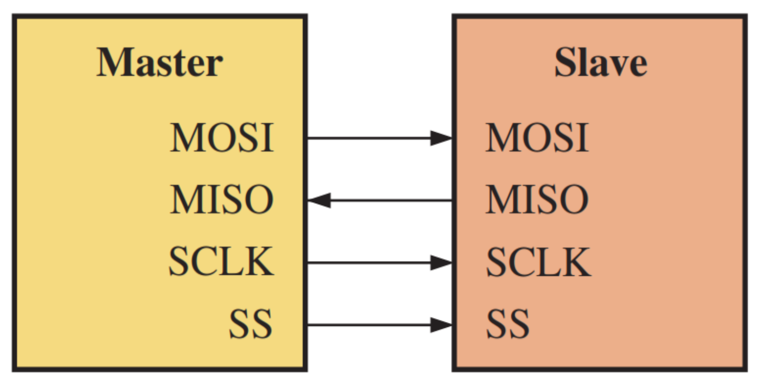


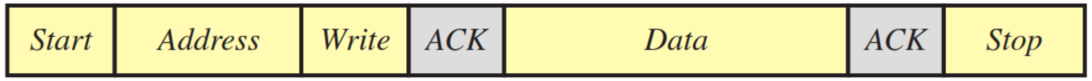
FIGURE 8: SPI master/slave configuration.

The I^2C Bus

- I^2C bus (pronounced I squared C) is also stated at I2C (I two C) and stands for inter- integrated circuit. It is an internal serial bus primarily for connecting ICs on a PCB.
- A main advantage is that it requires only two lines (plus ground) and therefore saves considerable board space compared to a parallel bus.
- Two signals (SDA and SCL) are used to communicate between compatible devices. Data are sent serially on the SDA line, and a clock is sent on the SCL line.
- Like, SPI I^2C is based on the master/slave concept where the master device drives the clock line and the slaves respond to the master.
- Only the master can initiate a transfer over the bus, but slaves can transfer data under control of the master using clock rates up to 100 kHz in the standard mode.
- Two other modes, enhanced and high- speed, allow 400 kbps and 3.4 Mbps, respectively.

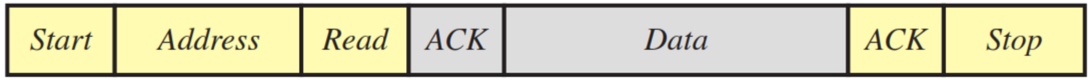
The I^2C Bus

- ☐ When transferring data from master to slave, the master device sends a start bit, followed by a slave address, and a write bit.
- ☐ The master waits for an acknowledge (ACK) signal from the slave and then sends the data and waits for an acknowledge before sending a stop bit, as illustrated in Figure 9(a). The yellow segments are from the master and the gray elements are from the slave.
- ☐ Similarly, when the master requires data from the slave, it sends a start bit followed by the address and a read bit.
- ☐ The slave returns an acknowledge followed by the data. When the master receives the data, it issues an acknowledge and a stop bit, as shown in Figure 9(b).



Start	*Address*	*Write*	*ACK*	*Data*	*ACK*	*Stop*

(a) Data transfer from master to slave



Start	*Address*	*Read*	*ACK*	*Data*	*ACK*	*Stop*

(b) Data transfer from slave to master

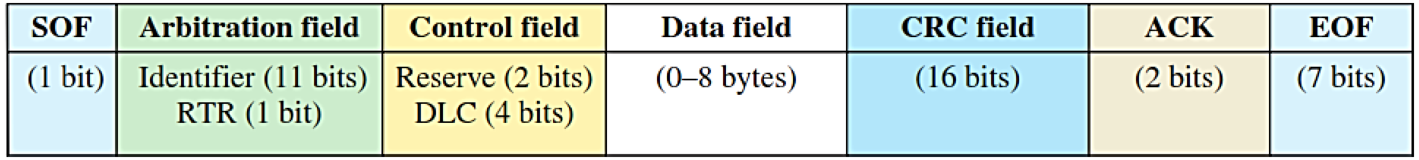
FIGURE 9: I^2C data transfers. Yellow is from master. Gray is from slave.

The CAN Bus

- The controller area network (CAN) bus, a differential serial bus, was developed for automotive applications and is also commonly used in aerospace systems, as well as other applications.
- Devices, called nodes, can be connected to the bus but are not assigned specific addresses as in the I^2C.
- The bus consists of a terminated twisted pair of signal lines, called CAN H and CAN L, plus ground.
- Two CAN specifications are in use. The standard or basic CAN 2.0A has 11-bit message identifiers and can operate up to 250 kbps, and the full CAN has 29-bit message identifiers and can be used up to 1 Mbps.
- The message identifier is a label for the contents of a message and goes to each node on the bus.

The CAN Bus

- Each receiving node performs a test on the identifier to determine if it is relevant to that node and is used to arbitrate the bus to determine if the message is of highest priority.
- All of the nodes on the bus can transmit and receive messages. The bus is available to a node with a message with the highest priority (dominant) and can override a message with lower priority (recessive).
- When the dominant message has been processed, the recessive message is retransmitted.
- The standard CAN data frame is shown in Figure 10. Data are transmitted in NRZ format.
- The frame begins with a start-of-frame (SOF) bit followed by an arbitration field and a control field.



SOF	Arbitration field	Control field	Data field	CRC field	ACK	EOF
(1 bit)	Identifier (11 bits) RTR (1 bit)	Reserve (2 bits) DLC (4 bits)	(0–8 bytes)	(16 bits)	(2 bits)	(7 bits)

FIGURE 10: Standard CAN data frame format.

The CAN Bus

- The arbitration field contains the message identifier and a remote transmission request (RTR) bit.
- The control field has two reserve bits and a data length code (DLC) that specifies the length of the data field that follows and can contain up to 8 bytes.
- The cyclic redundancy check (CRC) field provides for error detection. The acknowledge (ACK) verifies the receipt of correct data, and the frame ends with the end-of-frame field (EOF).
- An automobile typically has many control units (usually several dozen) for various sub-systems, including the engine control unit and other control units for transmission; ABS; cruise control; power steering; audio system; window, door, and mirror controls; airbags; and others.

The CAN Bus

- ☐ Figure 11 is a block diagram of a partial automotive control system using two CAN buses, one low-speed and one high-speed to control various functions throughout the vehicle.
- ☐ Each unit connected to the bus contains sensors and other functions that allow it to carry out its unique purpose.
- ☐ For example, the ABS (antilock braking system) can receive a message from sensors in each wheel indicating that the brake is about to lock up.

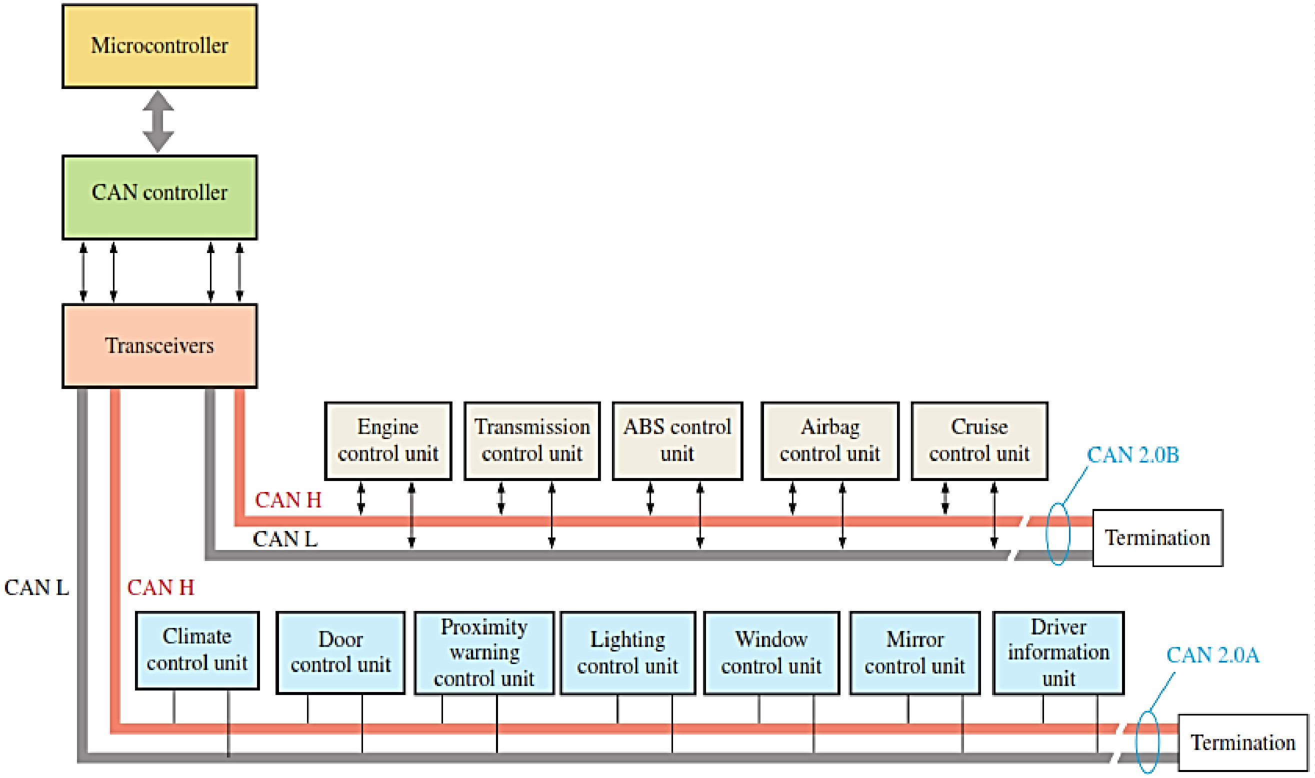


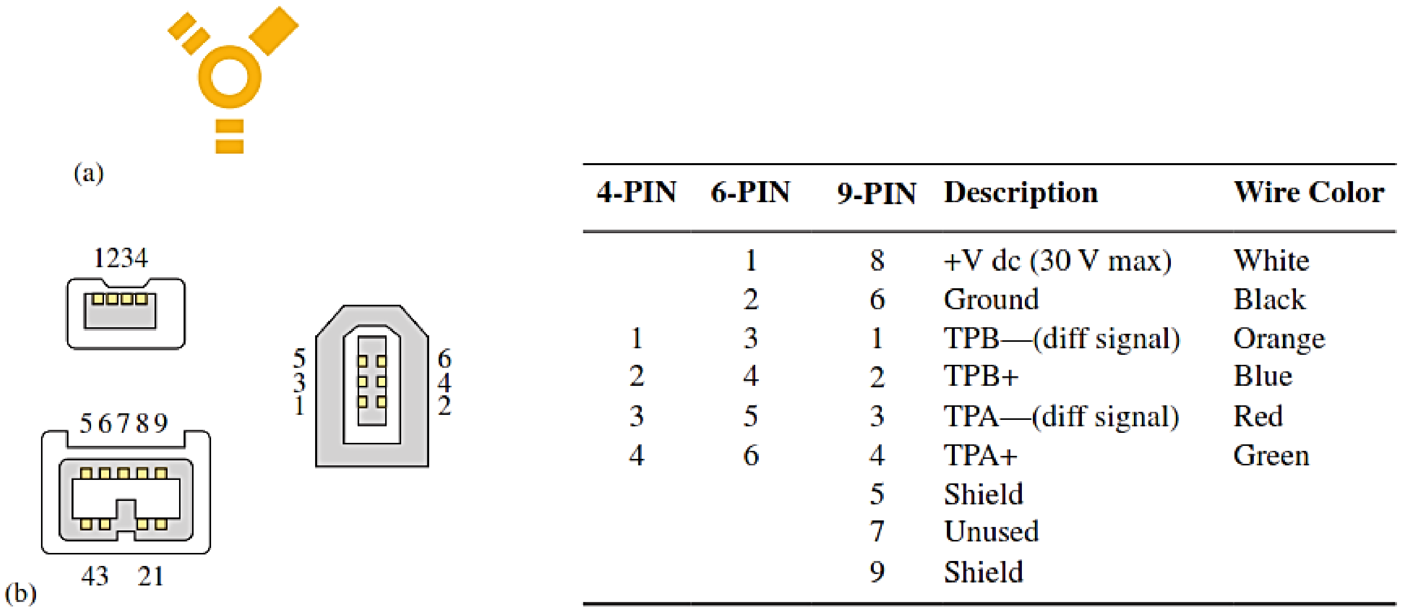
FIGURE 11: The CAN bus in an automotive control system.

The Firewire Bus

- Firewire, also known as IEEE-1394 and iLink, is a high-speed external serial bus developed by Apple Inc.
- Firewire is used in high-speed communications and real-time data transfer. It is used in professional audio and video equipment, camcorders, DVD players, external hard drives, and in computers used for audio and video editing, as well as in some auto and aircraft applications.
- It is similar to the USB except that it has a higher data rate and can handle more data.
- Three types of connectors are used in the Firewire standard: a 4-pin connector, a 6-pin connector, and a 9-pin connector.
- The cable for the 4-pin connector consists of two twisted pairs that carry data.
- The cable for the 6-pin connector has the two twisted pairs for data plus a power line and a ground line.

The Firewire Bus

- The cable for the 9-pin connector has the same wires as the 6-pin configuration plus two wires that provide for a grounded shield and one wire that is currently unused.
- The Firewire symbol is shown in Figure 12(a). End views of the three connector types are shown in part (b),
- and the pin designations are shown in part (c).
- The Firewire bus address has a total of 64 bits. Ten are for bus ID, six are for node ID, and 48 are for individual addresses. This allows up to 1023 buses, each having up to 63 nodes.



4-PIN	6-PIN	9-PIN	Description	Wire Color
	1	8	+V dc (30 V max)	White
	2	6	Ground	Black
1	3	1	TPB—(diff signal)	Orange
2	4	2	TPB+	Blue
3	5	3	TPA—(diff signal)	Red
4	6	4	TPA+	Green
		5	Shield	
		7	Unused	
		9	Shield	

FIGURE 12: Firewire symbol with cable and connector wires and pins.

Firewire versus USB

- In general, any capable node can control the bus in a Firewire system, but a single host is used to control the bus in USB.
- USB networks use a tiered-star topology and Firewire uses a tree topology. A Firewire device can communicate with any node at any time if the conditions allow, but a USB 2.0 device cannot communicate with the host device unless requested by the host.
- However, USB 3.0 allows Firewire-like communications between devices. USB provides 5 V power while Firewire provides up to 30 V.
- As a result, Firewire can supply more power to a device than USB. As mentioned before, Firewire is faster than USB.

Serial SCSI

- **Serial Attached SCSI** (SAS) is a data-transfer technology for transmitting data to and from storage devices.
- It has become a replacement for parallel SCSI bus technology that is commonly used in data centers, workstations, and servers.
- The serial SCSI overcomes some of the limitations of the parallel SCSI.
- The SAS supports up to a 12 Gbps data rate and allows up to 65,535 devices to be connected using expanders compared to 15 devices for parallel SCSI.

END of Lecture # 07